На правах рукописи

Семячкин Дмитрий Александрович

УПРАВЛЕНИЕ ПАРАЛЛЕЛЬНЫМИ ЗАДАНИЯМИ В ГРИДЕ С ПОМОЩЬЮ ОПЕРЕЖАЮЩЕГО ПЛАНИРОВАНИЯ

Специальность 05.13.11 — Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание учёной степени
кандидата физико-математических наук

Москва — 2008

Работа выполнена в Институте прикладной математики
им. М.В. Келдыша Российской академии наук.

Научный руководитель: доктор физико-математических наук, профессор **Корягин Дмитрий Александрович**.

Официальные оппоненты: доктор физико-математических наук, профессор **Крюков Виктор Алексеевич**,

кандидат физико-математических наук **Корухов Станислав Васильевич**.

Ведущая организация: Научно-исследовательский институт ядерной физики им. Д.В. Скобельцына Московского государственного университета им. М.В. Ломоносова (НИИЯФ МГУ).

Защита состоится 16 декабря 2008 г. в 11 часов на заседании Диссертационного совета Д 002.024.01 в Институте прикладной математики им. М.В. Келдыша РАН по адресу: 125047, Москва, Миусская пл., 4.

С диссертацией можно ознакомиться в библиотеке Института прикладной математики им. М.В. Келдыша РАН.

Автореферат разослан «____» _____ 2008 г.

Учёный секретарь диссертационного совета,

доктор физико-математических наук Т.А. Полилова

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы

В последнее время активное развитие получила новая модель организации ресурсов под названием грид — пространственно распределённая среда, интегрирующая множество ресурсов разных типов (процессоры, долговременная и оперативная память, хранилища файлов, базы данных и сети), совокупность которых может быть использована для решения прикладных задач нового уровня сложности. Потенциал технологий грида уже сейчас оценивается очень высоко: он имеет стратегический характер, и в близкой перспективе грид должен стать инструментарием для развития высоких технологий в различных сферах человеческой деятельности.

Наиболее развит и востребован на практике рассматриваемый в работе вычислительный грид, оперирующий такими типами ресурсов, как процессоры, оперативная и дисковая память, которые применяются для обработки заданий и хранения файлов. К настоящему времени уже разработаны ключевые для этого типа грида протоколы дистанционного запуска заданий и управления файлами.

Эффективность функционирования грида как среды с коллективной формой обслуживания пользователей определяется в первую очередь согласованностью распределения имеющихся ресурсов, которое должно происходить автоматически, опираясь на планирование вычислительных процессов в гриде в целом. Поэтому одной из ключевых функций, требуемых от программного обеспечения грида, является функция диспетчеризации, с помощью которой обеспечивается распределение ресурсов из общего ресурсного пула между заданиями, доставка программ и данных. Задача диспетчеризации много раз успешно решалась для ближайшего аналога вычислительного грида — кластерных систем, однако в условиях грида она значительно усложняется, и для её решения требуются новые подходы.

В архитектуре грида функция диспетчеризации реализуется специальными программными службами, обеспечивающими такой уровень интеграции распределённых ресурсов, при котором грид представляется в виде единой операционной среды обработки запросов (заданий). Совокупность таких служб составляют систему диспетчеризации. Большинство существующих на сегодня систем диспетчеризации предназначено для обслуживания гридов, состоящих из кластеров, — традиционной формы организации распределённых ресурсов. Используемым на практике системам диспетчеризации присущи довольно жёсткие ограничения по применению, и они не способны исключить такие нежелательные эффекты, как непредсказуемость времени обработки заданий, задержка обработки в ситуациях, когда имеются простаивающие ресурсы. Существенным недостатком большинства систем является невозможность

обработки параллельных заданий. Основная сложность в этом случае состоит в необходимости планирования, которое обеспечивает накопление и затем гарантированное синхронное выделение ресурсов в нескольких кластерах (коаллокация ресурсов): это предотвращает зависание заданий, которое является следствием фрагментации ресурсного пула. Некоторые системы способны решить эту задачу в специальных условиях применения, когда используемые ресурсы полностью отчуждаются в грид и централизованно управляются.

Диссертационная работа посвящена проблемам разработки методов управления параллельными заданиями и их алгоритмической поддержки для актуальной формы грида, когда ресурсы не отчуждаются от владельцев, а используются в гриде совместно с ними (неотчуждаемые ресурсы). Решение задачи в такой постановке открывает возможность создания высокопроизводительных вычислительных комплексов посредством интеграции пространственно распределённых, автономно управляемых, не выделенных специально в грид многопроцессорных и кластерных систем в единую операционную среду и применения в качестве средства межпроцессорного обмена данными глобальных телекоммуникаций.

Цель и задачи работы

Целью диссертационной работы является разработка нового метода управления параллельными заданиями в гриде. Достижение цели связывается с решением следующих задач.

Первая задача — это исследование существующих методов управления параллельными заданиями в кластерных системах и различных формах грида.

Вторая задача состоит в формализации планирования параллельных заданий для следующей формы грида:

- ресурсы используются совместно с владельцами (неотчуждаемые ресурсы);
- ресурсы организованы в кластеры (кластеризованные ресурсы);
- объекты планирования — многопроцессорные (параллельные) задания.

Третья задача — разработка архитектуры системы диспетчеризации для этой формы грида.

Четвёртая задача — разработка алгоритма планирования, решающего задачу коаллокации в условиях разделения кластерных ресурсов с их владельцами.

Пятая задача заключается в программной реализации разработанного метода в системе диспетчеризации параллельных заданий и оценке характеристик масштабируемости системы и эффективности алгоритма планирования.

Научная новизна

В диссертации разработан новый метод управления параллельными заданиями, позволяющий обеспечивать эффективное распределение ресурсов между разными пользователями и управлять выделением ресурсов для актуальной формы грида с неотчуждаемыми кластеризованными ресурсами. Предложен подход к диспетчеризации, опирающийся на моделирование загрузки кластерных ресурсов и составление расписания запуска заданий, в котором учитывается регулируемая стоимость ресурсов и приоритетность заданий. В рамках этого подхода разработан оригинальный алгоритм планирования, решающий задачу коаллокации ресурсов грида и способный подбирать ресурсы по критериям скорейшего старта или скорейшего завершения задания. В программной реализации прототипа системы диспетчеризации параллельных заданий применены новые для диспетчеризации в гриде механизмы резервирования и предсказания загрузки кластерных ресурсов.

Практическая значимость

Полученные в диссертационной работе результаты могут быть использованы для построения гридов из существующих вычислительных центров путём объединения их ресурсов для решения важных прикладных задач науки и техники, выполняющихся на большом числе процессоров.

Разработанный диспетчер позволяет повысить эффективность функционирования распределённой вычислительной среды, а работу с ней сделать не сложнее, чем с более привычными компьютерными архитектурами: многопроцессорными и кластерными системами. С его помощью можно решать наиболее ресурсоёмкие параллельные задачи, для которых требуется привлечение компьютерного парка нескольких организаций.

Предполагается, что в дальнейшем результаты работы будут применены в программном обеспечении крупных инфраструктурных проектов.

Апробация работы

Основные результаты работы докладывались и обсуждались на следующих конференциях и семинарах:

1. 1-я международная конференция «Распределённые вычисления и грид-технологии в науке и образовании». Доклад «Использование алгоритма Backfill в гриде», Дубна, 29 июня-2 июля 2004 г.
2. Семинар МГУ им. М.В. Ломоносова «Проблемы современных информационно-вычислительных систем» под руководством д.ф.-м.н. В.А. Васенина. Доклад «Способы планирования в гриде и их реализация в грид-диспетчере», Москва, 12 апреля 2005 г.

3. Семинар группы разработчиков программного обеспечения для грид-инфраструктуры EGEE ARDA под руководством M. Lamanna. Доклад «KIAM in GT4 Evaluation Activity and Grid Research», CERN, Женева, 12 октября 2005 г.
4. 13-я Всероссийская научно-методическая конференция «Телематика-2006». Доклад «Создание прототипа центра базовых грид-сервисов нового поколения для интенсивных операций с распределёнными данными в федеральном масштабе», Санкт-Петербург, 5-8 июня 2006 г.
5. 2-я международная конференция «Распределённые вычисления и грид-технологии в науке и образовании». Доклад «Управление параллельными заданиями в гриде с помощью метода опережающего планирования», Дубна, 26-30 июня 2006 г.
6. Научная конференция «Ломоносовские чтения», факультет ВМиК МГУ им. М.В. Ломоносова. Доклад «Коаллокация ресурсов грида для обслуживания параллельных заданий», Москва, 16-24 апреля 2008 г.
7. Научный семинар ИПМ им. М.В. Келдыша под руководством М.Р. Шура-Бура и Д.А Корягина. Доклад «Управление параллельными заданиями в гриде с помощью опережающего планирования», Москва, 6 ноября 2008 г.

По материалам диссертации опубликовано пять печатных работ [1, 2, 3, 4, 5], в том числе, одна [5] — в журнале, рекомендованном ВАК для публикации основных результатов докторских и кандидатских диссертаций по вычислительной технике и информатике.

Структура и объём работы

Работа состоит из введения, четырёх глав, заключения и списка литературы. Общий объём диссертации — 114 страниц (включая 10 страниц приложения). Список литературы содержит 61 наименование. В работе содержится 11 рисунков и 3 таблицы.

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность и практическая значимость диссертации, рассматриваются цель и задачи исследования, а также излагается краткое содержание диссертационной работы.

Первая глава содержит изложение понятий, методов и программных средств, лежащих в основе разработок, которые описываются в следующих главах диссертационной работы. Глава состоит из трёх частей.

В первой части определяется параллельное задание как программа, реализованная в виде нескольких компонент, каждая из которых запускается на отдельном процессорном узле и в ходе своего выполнения может взаимодействовать с остальными компонентами, а также вводится понятие планирования как основного этапа управления параллельными заданиями.

Даётся обзор двух групп методов планирования, используемых в современных параллельных архитектурах и кластерных системах. Методы первой группы используют разделение времени процессоров между несколькими заданиями. Отмечается основной недостаток этих методов, состоящий в том, что компоненты одного задания могут быть прерваны и перезапущены в различные моменты времени, что приводит к сильному снижению общей эффективности использования ресурсов. Вторая группа методов планирования (FCFS, First-Fit, Backfill) основана на идее разделения пространства ресурсов между заданиями, согласно которой каждое задание получает необходимый объём ресурсов на требуемое время в эксклюзивном режиме. Подробно рассматривается широко применяющийся на практике метод обратного заполнения Backfill, гарантирующий запуск параллельного задания и вместе с тем эффективно использующий ресурсы благодаря выделению ресурсов заданиям не непосредственно в момент освобождения, а заблаговременно.

Во второй части обсуждаются технологии распределённого компьютинга, в частности, концепция и базовые принципы грида как способа организации инфраструктуры высокопроизводительного компьютинга. Современный и наиболее распространённый подход к такой организации состоит в построении грида из пространственно распределённых многопроцессорных установок, таких как массивно-параллельные компьютеры (MPP) и кластеры. Отмечается, что в существующих проектах (EGEE, Grid2003, NorduGrid и др.) преобладает способ создания грида, когда сами владельцы не используют свои ресурсы, а отдают их целиком в грид. Такой способ организации ресурсов, называемый гридом с отчуждаемыми ресурсами, не может претендовать на универсальность, однако может быть полезен в специальных условиях применения, например, когда владельцы сами не используют свои ресурсы, а выступают в роли провайдеров. На практике представляется более интересным способ, при котором ресурсы используются совместно их владельцами и пользователями грида. Этот способ организации называют гридом с неотчуждаемыми ресурсами. Его достоинство состоит в том, что такой грид можно создать лишь на некоторое время, требуемое для решения конкретной задачи, и без формирования специальной ресурсной базы: достаточно лишь вовлечь уже имеющиеся ресурсы, не полностью загружаемые своими владельцами.

В третьей части рассматривается специфика управления параллельными заданиями в гриде по сравнению с параллельными компьютерными архитектурами и кластерными системами. Приводятся наиболее значимые отличительные свойства грида, усложняющие задачу коаллокации: автономность, гетерогенность, пространственная распределённость и двухуровневая организация ресурсов. Ставится задача коаллокации — центральная задача планирования параллельных заданий, состоящая в размещении набора компонент параллельного задания на множестве ресурсов. Для условий грида она формулируется следующим образом. В качестве ресурсов рассматриваются вычислительные ресурсы

грида — кластеры $R = \{R_1, R_2, ..., R_N\}$, каждый из которых представляет собой некоторое количество процессорных узлов — процессоров и связанную с каждым из них оперативную и дисковую память. На эти ресурсы необходимо разместить P компонент параллельного задания $J = \{J_1, J_2, ..., J_P\}$, синхронно выделяя им ресурсы из множества ресурсов R на время выполнения T.

В результате решения задачи определяется время старта задания и набор аллокаций ресурсов: $A = \{A_1(r_{j_1}, t_0, t_0 + T), ..., A_K(r_{j_K}, t_0, t_0 + T)\}, K = \overline{1, P}$, где r_{j_i} — ресурсы, используемые для аллокации такие, что: $r_{j_i} \subseteq R_{j_i}, i = \overline{1, K}$, $|r_{j_1}| + |r_{j_2}| + ... + |r_{j_K}| = P$, где $|r_{j_i}|$ — количество аллоцированных процессорных узлов на ресурсе. Все аллокации начинаются в одно время и имеют одинаковую длительность. Кроме того, обеспечивается выполнение двух условий:

- характеристики исполнительных ресурсов отвечают требованиям компонент задания;
- исполнительные ресурсы в период $[t_0, t_0 + T]$ свободны, и политика разделения ресурсов не препятствует их выделению для задания.

Перечисленные выше свойства грида делают возможность построения точных аллокаций $A_1, ..., A_K, K = \overline{1, P}$, в которых определяется множество исполнительных процессоров $r_{j_1}, ..., r_{j_K}$ и временной интервал $[t_0, t_0 + T]$, на который они отводятся этому заданию, проблематичной. Вместо этого во многих практических реализациях результатом планирования являются аллокации, в которых время начала и ресурсы точно не определены. Для параллельных заданий это влечёт за собой серьёзные дефекты при их обработке, такие как «зависание» заданий в очереди кластера.

Делается вывод о том, что для предотвращения нежелательных эффектов планирование в гриде должно быть детерминированным, то есть должно строить точные аллокации, и, кроме того, запуск заданий на исполнительных ресурсах должен осуществляться в соответствии с построенными аллокациями.

Вторая глава посвящена рассмотрению подходов к планированию параллельных заданий в гриде на примере существующих программных разработок. Глава состоит их двух частей.

В первой части даётся обоснование выбора в пользу приоритетного планирования для применения в гриде и рассматривается классификация методов такого планирования.

Как показывает практика, методы разделения времени трудно применимы в условиях грида. Кроме того, эти методы направлены на улучшение среднего времени обработки заданий, тогда как в гриде планирование в первую очередь должно иметь целью обеспечение качества обслуживания отдельных пользователей. Для этого более подходят методы разделения пространства ресурсов, которые в настоящее время преимущественно используются в гриде. Среди них выделяются методы приоритетного планирования и методы, основанные на интегральных

критериях. Последние способны оптимизировать загрузку ресурсов или общее время счёта пакета заданий. Представляется, что планирование в гриде в первую очередь должно быть направлено на справедливое разделение ресурсов между заданиями, обеспечивая при этом возможность управления порядком выделения ресурсов для отдельных заданий. Для таких целей наиболее подходящим является приоритетное планирование.

Способ решения задачи коаллокации зависит в значительной степени от того, какая информация о ресурсах для планирования имеется в наличии. По этому критерию современные методы приоритетного планирования разделяются на два класса. К первому классу принадлежат методы, основанные на использовании очередей заданий, основной принцип которых заключается в выделении ресурсов для стоящих в очереди заданий, исходя из текущего состояния ресурсов. Применение этих методов для параллельных заданий в гриде неэффективно по двум причинам. Во-первых, накопление ресурсов, необходимых для запуска задания, может быть осуществлено лишь с помощью блокировки свободных ресурсов, что приводит к их простою в течение неопределённого времени. Во-вторых, механизм предварительного резервирования, предназначенный для обеспечения детерминированного планирования, трудно применим из-за отсутствия информации о времени освобождения требуемого заданию объёма ресурсов.

Второй класс включает методы, использующие для распределения ресурсов полноценный план на будущее или расписание. К ним относится рассмотренный в первой главе диссертации метод обратного заполнения (Backfill). С их помощью автоматически обеспечивается накопление ресурсов и естественным образом реализуется механизм предварительного резервирования ресурсов, что позволяет использовать методы этого класса для обслуживания параллельных заданий в гриде.

Во второй части анализируются известные системы планирования параллельных заданий в гриде (KOALA, MSS, JSS, NWIRE, CCS). Особое внимание уделяется способу решения в них задачи коаллокации.

В **третьей главе** описан предложенный автором метод управления параллельными заданиями с помощью опережающего планирования для практически важной формы организации грида с неотчуждаемыми ресурсами, когда они не выделяются в грид полностью, а используются в режиме разделения с их владельцами. В этих условиях задания поступают не только из грида (глобальный уровень), но и от пользователей кластера (локальный уровень) (рис. 1). Особенностью предложенного решения является то, что ресурсы распределяются не в момент их освобождения, а строится расписание их использования на некоторый период времени в виде множества временных слотов, где каждый слот соответствует началу и концу некоторого задания или свободному участку расписания. Выполнение расписания обеспечивается с помощью предварительного резервирования ресурсов. Благодаря этому планирование в условиях неотчуждаемости ресурсов является детерминированным.

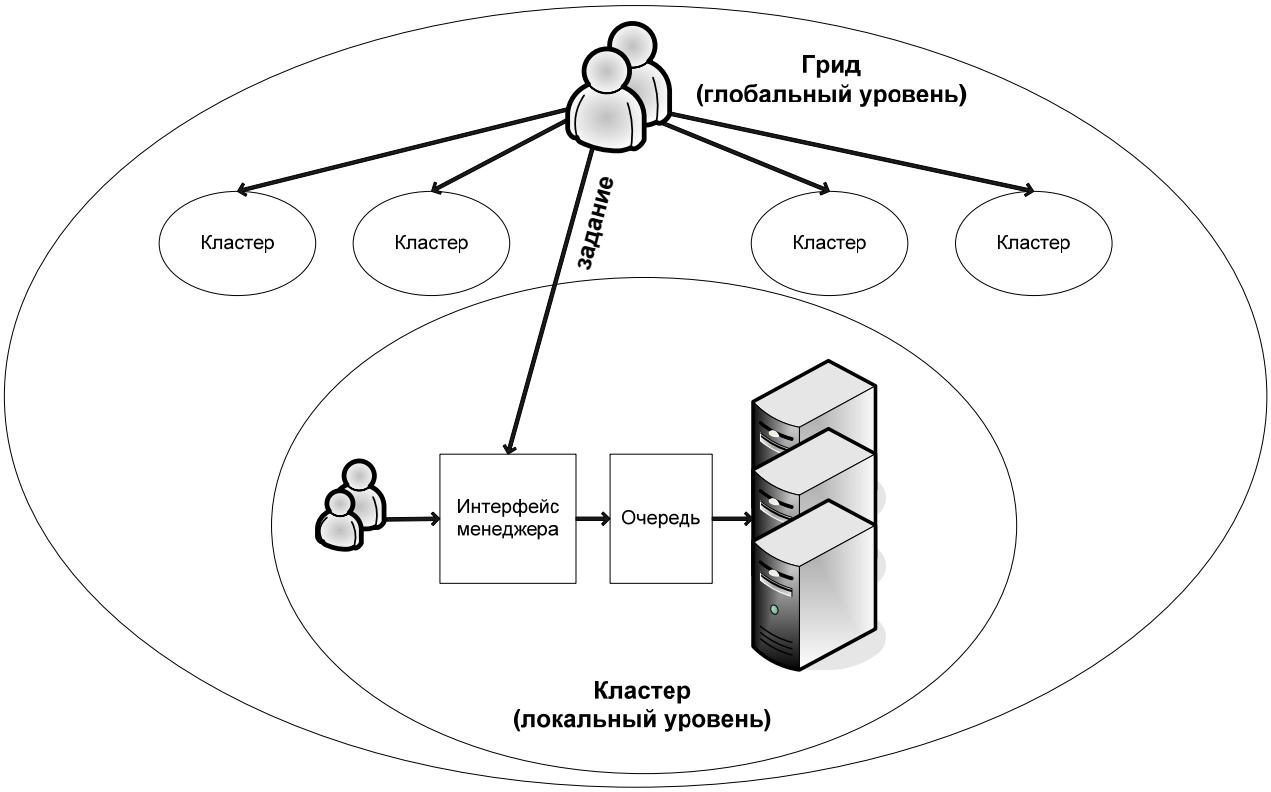

Рис. 1. Грид с неотчуждаемыми ресурсами.

В первой части рассматривается экономический подход к планированию в гриде для управления параллельными заданиями. Согласно этому подходу процесс распределения ресурсов организуется на базе модели рынка: владельцы ресурсов выступают в качестве продавцов, а пользователи — в качестве покупателей этих ресурсов. Реализуя такой подход, предлагаются два соглашения по разделению ресурсов и способы их поддержки при планировании. Первое соглашение касается разделения ресурсов между пользователями. Основное требование заключается в том, что пользователь должен иметь возможность управлять скоростью получения ресурсов при планировании. Для этого им устанавливается плата за выполнение каждого задания в рамках выделенного ему бюджета. При увеличении платы шансы быстрее выполнить задание повышаются, однако при этом ограничение бюджета не позволяет пользователю монополизировать весь ресурсный пул. Для разделения ресурсов между двумя потоками заданий, поступающих с локального и глобального уровней, предлагается второе соглашение о разделении ресурсов: глобальное задание может занять ресурсы при условии, что плата, назначенная пользователем за выполнение задания, не меньше их стоимости. Такое соглашение позволяет осуществлять динамическую балансировку потоков заданий, обеспечивая конкуренцию глобальных и локальных заданий, и является ключевым для грида с неотчуждаемыми ресурсами.

Согласно этому подходу глобальные задания могут размещаться не только на свободных ресурсах, но и на ресурсах, на которые претендуют локальные задания, при условии, что плата за эти задания не ниже стоимости

ресурсов. Таким образом, планирование параллельных заданий заключается в подборе множества «подходящих» слотов, которые:

- можно синхронно выделить в количестве, необходимом заданию;
- удовлетворяют ресурсному запросу задания;
- подходят по стоимости.

Во второй части приводится алгоритм планирования, решающий задачу коаллокации ресурсов с учётом соглашений по их разделению, а также его обоснование и экспериментальная оценка, проведённая на разработанном прототипе диспетчера заданий. Показывается, что один из лучших на сегодняшний день методов планирования параллельных заданий — метод обратного заполнения (Backfill) — в условиях неотчуждаемости ресурсов имеет квадратичную сложность от общего количества слотов. Для этих условий в работе предлагается алгоритм планирования линейной сложности и его вариант, реализующий критерий скорейшего завершения заданий.

Алгоритм рассчитан на следующие условия.

- Все компоненты параллельного задания имеют одни и те же требования к ресурсам компьютера, выраженные в ресурсном запросе.
- Плата за каждую компоненту задания составляет RC/P, где P — количество компонент, а RC — стоимость задания в целом. Таким образом, в соответствии с принципом разделения ресурсов, заданию могут быть выделены только те из них, которые имеют стоимость, меньшую RC/P.

Алгоритм подбирает слоты для одного задания в два этапа и основан на совместном использовании двух представлений расписания в виде списков L_1 и L_2, в первом из которых слоты отсортированы по возрастанию времени начала, а во втором — по возрастанию времени конца. Эта сортировка выполняется один раз для всей очереди заданий перед началом работы алгоритма.

На первом этапе алгоритма подсчитывается количество подходящих слотов в каждый момент времени. Изменение этого числа происходит только в точках временной оси, соответствующих началу или концу какого-либо слота. Для подсчёта выполняется параллельный проход по спискам L_1 и L_2. При переходе к новому слоту из списка L_1, время начала аллокации t устанавливается равным началу этого слота. Осуществляется проверка, является ли слот подходящим для запуска задания. Если это так, то количество подходящих слотов увеличивается. После этого проходятся слоты второго списка, конец которых меньше $t+T$, и соответственно уменьшается количество подходящих слотов. Как только набирается необходимое количество подходящих слотов P, этот этап работы алгоритма завершается, и получается время $t_0 = t$, начиная с которого для задания можно синхронно выделить требуемое количество слотов.

Если на первом этапе необходимое количество слотов найдено, то все слоты, лежащие на интервале $[t_0, t_0 + T]$ и являющиеся подходящими, могут

быть использованы для его запуска. Для их отбора совершается ещё один проход по первому списку L_1 (второй этап). Полученный набор слотов (или их частей) является искомым набором аллокаций.

Представленный алгоритм определяет самое раннее время старта задания. Если в системе имеются процессорные узлы с разной производительностью, то более полезным является критерий скорейшего завершения задания. Для реализации этого критерия предлагается модификация алгоритма, которая пытается в первую очередь подобрать слоты на наиболее производительных процессорах, чтобы выполнить задание как можно быстрее.

Для анализа эффективности разработанного алгоритма планирования реализован планировщик, рассматриваемый в четвёртой главе, и подготовлена экспериментальная среда, основанная на событийном моделировании. С её помощью получены показатели, используемые для оценки эффективности алгоритмов планирования: среднее время ожидания, время счёта пакета заданий и степень загрузки ресурсов, а также построена зависимость среднего времени планирования от количества заданий в системе. На основе полученных результатов проведено сравнение разработанного алгоритма планирования с алгоритмом FCFS в условиях грида с неотчуждаемыми ресурсами, показавшее существенно более высокие показатели эффективности планирования первого алгоритма. Также исследовано влияние изменения стоимости ресурсов на время отклика и масштабируемость планировщика.

В **четвёртой главе** рассматривается программная реализация прототипа системы диспетчеризации параллельных заданий, в основу которой положены разработанные в диссертационной работе метод и алгоритм. Система основывается на современном подходе к построению грид-систем — Открытой архитектуре грид-служб (OGSA), реализуемой с помощью концепции веб-служб (Web Service Architecture — WSA) и соответствующих стандартных протоколов. В основу реализации положено промежуточное программное обеспечение, признанное стандартом де-факто, — инструментарий Globus Toolkit 4. Особенностью системы является то, что для планирования диспетчер использует прогноз использования ресурсов на будущее, что позволяет эффективно распределять задания по ресурсам.

Первая часть главы содержит подробное описание предлагаемой архитектуры системы, включающей в себя три основных компоненты: диспетчер, ресурсный агент и пользовательский интерфейс. На рис. 2 серым цветом выделены разработанные блоки, являющиеся частью системы, а компоненты, реализованные в виде веб-служб (WS-компоненты), заключены в жирную рамку. Каждой компоненте посвящён отдельный раздел.

Во второй части рассматривается диспетчер, представляемый набором грид-служб, реализующих базовые компоненты, необходимые для распределения заданий по ресурсам.

Состав диспетчера включает:

- службу приёма команд от пользователей;

- службу приёма информации о ресурсах;
- базу данных планирования;
- службу планирования;
- службу управления запуском, включающую в себя менеджер резервирования, менеджер запуска и менеджер доставки данных.

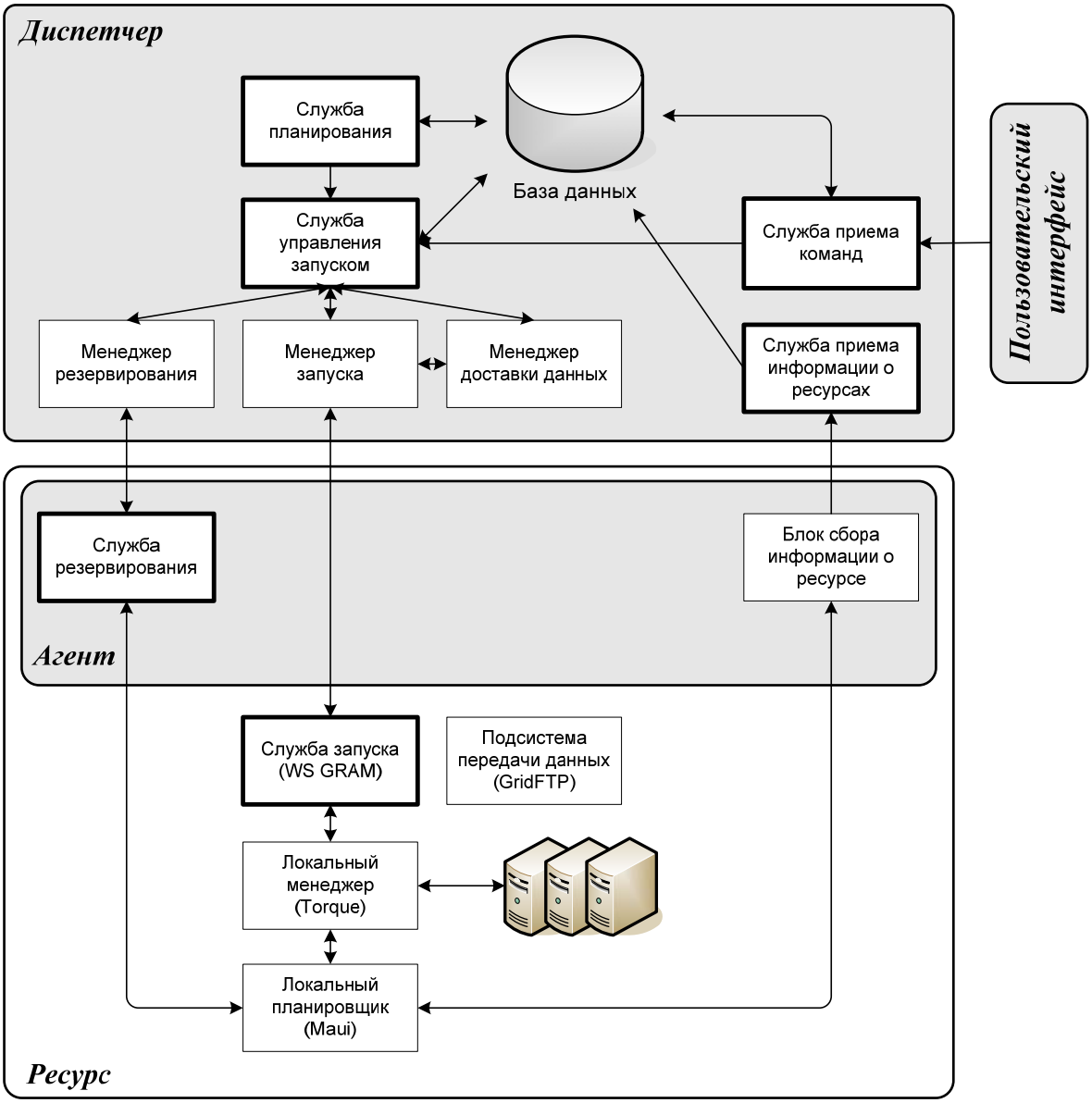

Рис. 2. Архитектура системы диспетчеризации.

Работа службы планирования — основной компоненты диспетчера — является циклической: на каждом цикле строится план распределения заданий по ресурсам на основе информации, находящейся в базе данных планирования. Во время работы цикла все задания, поступающие от службы приёма заданий, и обновления кластерных расписаний, поставляемые ресурсными агентами, буферизуются средствами СУБД и не учитываются на текущем цикле. По окончанию цикла они актуализируются в базе данных для использования на следующих циклах. После построения плана определяется подмножество заданий, время начала аллокаций которых достаточно близко

к моменту начала выполнения. Для этих заданий посредством службы предварительного резервирования выполняется резервирование ресурсов, гарантирующее их выделение в соответствии с построенными планировщиком аллокациями, и инициируется доставка заданий в кластеры средствами Globus Toolkit.

В третьей части описывается ресурсный агент, основной функцией которого является сбор информации о состоянии ресурса, в частности, построение прогноза, и передача этой информации диспетчеру. Кроме того, на ресурсного агента возложена функция резервирования ресурсов, необходимая для обслуживания параллельных заданий. Реализация ресурсного агента, основана на специальном режиме SIMULATION кластерного планировщика Maui. Этот режим позволяет моделировать процесс размещения заданий в кластере в будущем времени, обеспечивая генерацию локальных расписаний. Для отслеживания изменения состояния кластера предложен способ, основанный на «прослушивании» сообщений, которые локальный менеджер посылает планировщику Maui. Таким образом, ресурсный агент узнаёт о событиях, происходящих в кластере, и управляет моделированием, в результате которого строится расписание. Также предложен метод динамического управления шагом планирования, позволяющий существенно сократить время построения прогноза. На основе средств Maui, позволяющих зарезервировать локальные ресурсы, разработан и задействован в диспетчере механизм предварительного резервирования ресурсов, в реализации которого решена важная задача закрепления резервирования за конкретным заданием грида, выполнение которого запланировано на соответствующих ресурсах.

В четвёртой части рассматривается пользовательский интерфейс, предоставляющий возможность запускать задание и получать информацию о его состоянии.

В **заключении** перечисляются основные результаты работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. На основе анализа существующих методов управления параллельными заданиями в многопроцессорных и кластерных системах показано, что в условиях грида эти методы не применимы непосредственно, а требуется их модификация, учитывающая особенности пространственно распределённой среды.
2. Предложен новый метод диспетчеризации параллельных заданий, позволяющий использовать ресурсы входящих в грид кластеров как пользователям грида, так и организациям, являющимся владельцами кластеров. Обеспечивается управляемость процессом обработки заданий со стороны пользователей, и в то же время контролируемость использования ресурсов владельцами кластеров.

3. Разработана архитектура системы диспетчеризации, в которой наряду с традиционно используемыми механизмами: очередью заданий, приоритетным управлением, дистанционной доставкой файлов — применены новые: механизмы предсказания загрузки кластерных ресурсов и их предварительного резервирования.
4. В рамках предложенной архитектуры решён ключевой вопрос управления параллельными заданиями: разработан оригинальный алгоритм планирования, решающий задачу коаллокации ресурсов грида и способный подбирать ресурсы по критериям скорейшего старта или скорейшего завершения задания.
5. Реализована система диспетчеризации параллельных заданий. Система установлена на экспериментальном полигоне грида, что позволило провести оценку характеристик масштабируемости системы и эффективности алгоритма планирования. Реализация может быть использована в грид-инфраструктурах, создание которых в России ожидается в ближайшей перспективе.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

[1]. Коваленко В.Н., Семячкин Д.А. Использование алгоритма Backfill в грид // Распределённые вычисления и Грид-технологии в науке и образовании: Труды международной конференции. Дубна: ОИЯИ, 2004. С. 139–144.

[2]. Коваленко В.Н., Семячкин Д.А. Управление параллельными заданиями в грид с помощью метода опережающего планирования // Распределённые вычисления и Грид-технологии в науке и образовании: Труды 2-й международной конференции. Дубна: ОИЯИ, 2006. С. 309–316.

[3]. Кореньков В.В., Березовский П.С., Галактионов В.В., Демичев А.П., Жильцов В.Е., Ильин В.А., Коваленко В.Н., Корягин Д.А., Крюков А.П., Мицын В.В., Семячкин Д.А., Стриж Т.А., Шамардин Л.В. Создание прототипа центра базовых GRID-сервисов нового поколения для интенсивных операций с распределёнными данными в федеральном масштабе // Телематика’2006: Тезисы докладов XIII Всероссийской научно-методической конференции. Санкт-Петербург: 2006.

[4]. Коваленко В.Н., Коваленко Е.И., Корягин Д.А., Семячкин Д.А. Управление параллельными заданиями в гриде с неотчуждаемыми ресурсами. Препринт № 63, 2007. М.: ИПМ РАН. 28 с.

[5]. Коваленко В.Н., Семячкин Д.А. Методы и алгоритмы управления параллельными заданиями в гриде с ресурсами в форме кластеров // Вестник Южного научного центра РАН. 2008. № 3(4). С. 23–34.

Подписано в печать 06.11.2008. Формат 60x90/16. Усл. печ. л. 1,0. Тираж 70 экз. Заказ 9-27.
ИПМ им.М.В.Келдыша РАН. 125047, Москва, Миусская пл., 4